MILS Real-time CORBA

OMG Real-time Workshop
July 2003

Jeff Chilton
Bill Beckwith
Objective Interface Systems
Herndon, Virginia

ORB*express*™

Agenda

- What is MILS?
- The Common Criteria
- Protection Profile
- Middleware Security Policies
- Threat Examples
- What About CORBASEC?
- The Path From Here

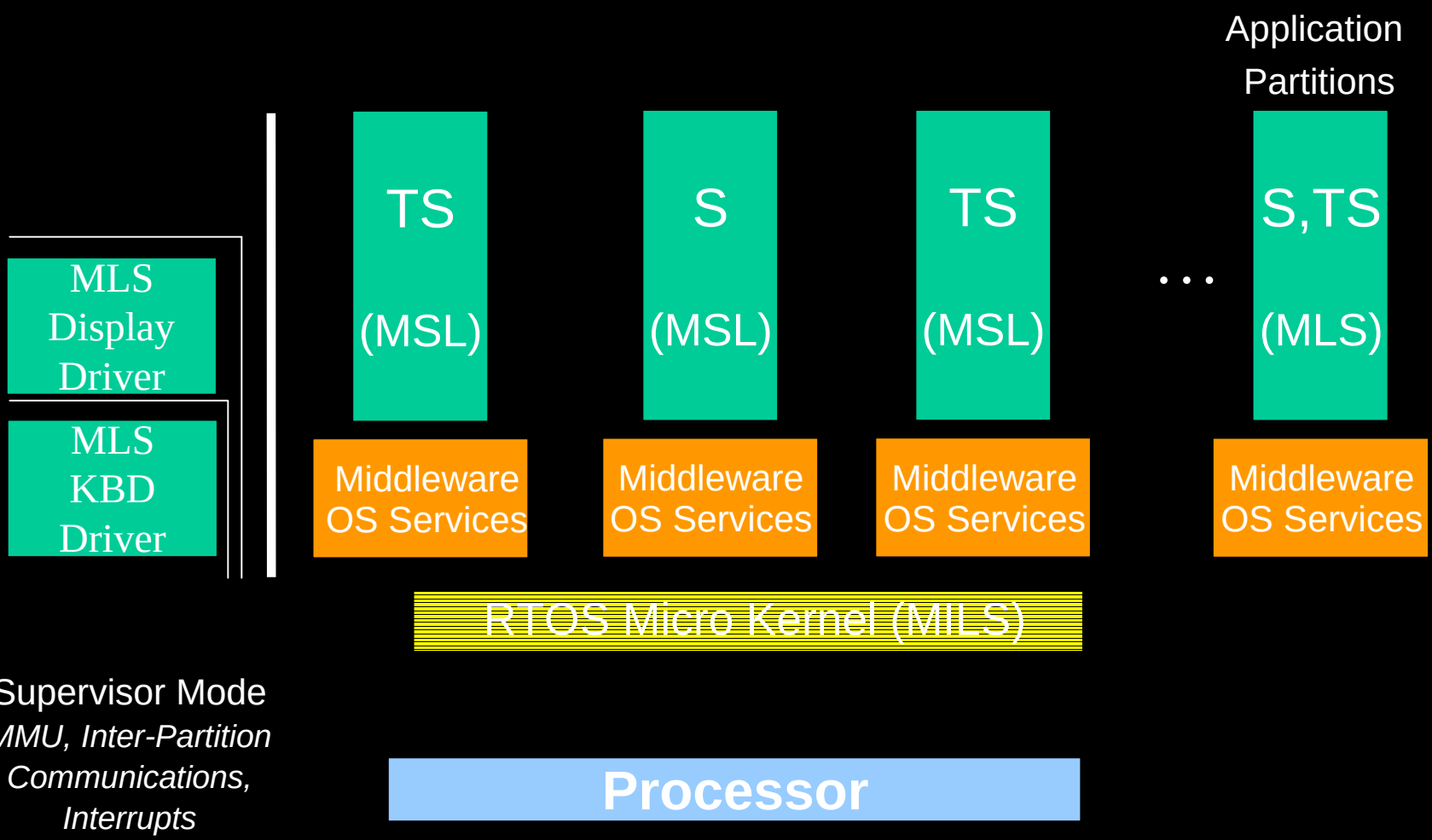
What is MILS ?

- **Multiple Independent Levels of Security**
- **An architecture for building High-Assurance systems**
  - Applies to both Multi-Level Security and Safety-Critical (DO-1780
- **The basic concept is to apply layering and separation of responsibility in an 'independent' fashion**

MILS Separation Architecture (Information Flow Paradigm)

- MILS Architecture utilizes a Layered Approach to Security
- Software Decomposed into 3 Layers (Rushby):
  - The Micro (Partitioning) kernel
  - The Middleware (e.g. ORB, System Control, drivers)
  - The (User) Application Software
- Partitioning Kernel
  - Creates separate Partitions and
  - Provides secure information flow (Pipes)
- Middleware Security Policy Enforcement provides for:
  - Application Component Creation and
  - Secure Inter-Object Message Flow
- Applications provide application specific Security Functions
  - Firewalls
  - Crypto Services
  - Web Servers)

MILS System



Layer Responsibilities

Partitioning Kernel Functionality

- Time and Space Partitioning
- Data Isolation
- Inter-partition Communication
- Periods Processing
- Minimum Interrupt Servicing
- Semaphores
- Timers
- Instrumentation

Middleware Functionality

- **RTOS Services**
- MILS Device Drivers
- Inter-processor Communication
- MILS CORBA
- Marshalling and Proxy Service
- MILS File System

MILS Middleware

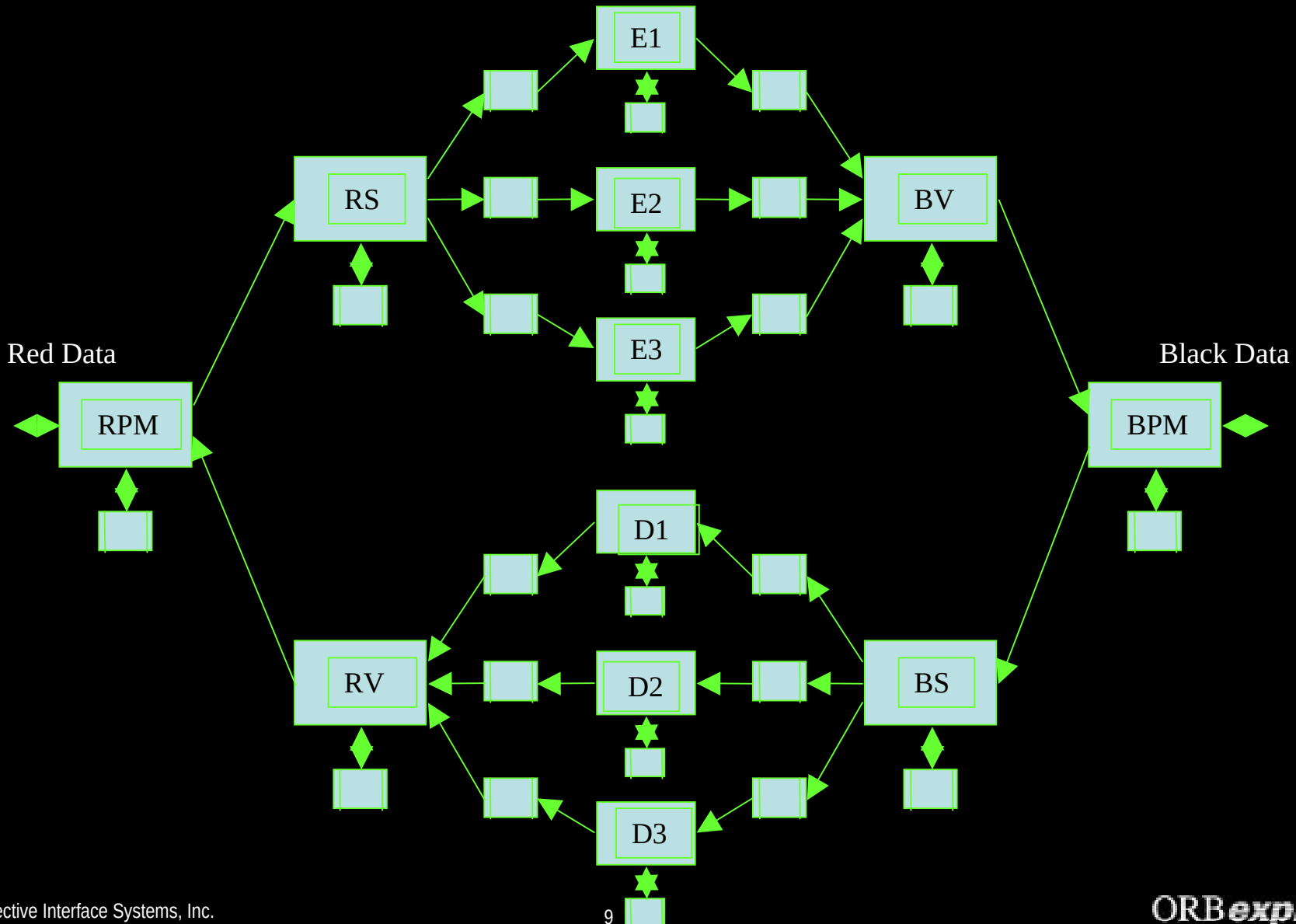
- Enforce security policy in a network centric manner :
  - End to End Information Flow
  - End to End Data Isolation
  - End to End Periods Processing
  - End to End Damage Limitation

- Multi-Level Secure (MLS) software must be (NEAT):
  - **N**on-Bypassable
  - **E**valuatable
  - **A**lways Invoked
  - **T**amperproof

Advantages of MILS

- **Manageable Evaluation**
  - Each layer may be evaluated separately without impact to the evaluation of the other layers.
- **Layering Improves Flexibility**
  - Higher assurance kernels may be incorporated without impact to evaluation of other layers
- **Extensible Security**
  - Supports new security enforcement mechanisms at the application level without impacting other enforcement mechanisms
- **High Assurance Applications:**
  - Can be developed and maintained
  - Can be evaluated

MILS Security Policy Application Example



The Common Criteria

- Result of a series of international efforts to develop criteria for evaluation of IT security
- Stems from US Trusted Computer System Evaluation Criteria (TCSEC) - circa 1985 (The Orange Book)
- Current version is 2.1 (and is now same as ISO 15408)

Who uses the Common Criteria?

- Consumers of IT technology - for guidance formulating requirements
- Developers - for reference formulating functional specifications
- Evaluators - as a mandatory statement of evaluation criteria

Protection Profile

- An implementation-independent statement of security requirements that is shown to address threats in a specified environment.
- Describes environment: assumptions, threats, security policies
- Lists security objectives
- Lists security function requirements
- Lists security assurance requirements
- Presents rationales (for mappings, every which way)

Security Policies for RT CORBA

- **Unauthorized Discloser of Data - >** Information Flow
  - Critical tasks not bypassed
- **Compromise/Corruption of Sensitive Data->**Data Isolation
  - Data segments not read or corrupted by unauthorized entities
- **Covert Storage Channels - >** Periods Processing
  - ORB is not a covert storage channel on separate messages
- **Presence of Unevaluated Code - >** Damage Limitation
  - Unevaluated code will not compromise processing or data

Threat Examples

- **T.CONFIG_CORRUPT**
  - A malicious or faulty subject may attempt to modify or corrupt configuration data used by the ORB to enforce information flow policy by accessing the contents of an ORB.
  - A malicious or faulty subject may attempt to bypass the information flow policy enforced by an ORB by using configuration data that, while valid, differs from that being used by another ORB.
- **T.DOS**
  - A malicious or faulty subject may attempt to block other subjects from sending or receiving communications by exhausting or monopolizing shared resources or the resources of another ORB.

What About OMG CORBASEC ?

- **CORBASEC Is Not Used**
  - ORBs reside in the application process space
    - Mechanisms can be modified by the application
  - Security mechanisms easily bypassed
    - Security becomes an application option
  - Collection of security objects
    - Rather than a security architecture
  - CORBASEC implementations too large to evaluate
  - Not intended for real-time and embedded
    - Developed for enterprise applications

The Path From Here

- **Common Criteria Protection Profile**
  - For Partition Kernel
  - For RT-CORBA
- **OMG High-Assurance RFP**

Contact Information

To Contact Objective Interface:

- Web Page
  http://www.ois.com/
  http://www.realtime-corba.com/
- E-mail
  info@ois.com
- Phone
  1-800-800-OIS7 (6477) or
  703/295-6500
- Fax
  703/295-6501
- Mail
  13873 Park Center Road,
  Suite 360
  Herndon, VA 20171-3247

"The great pleasure in life is doing what people say you cannot do."
— Walter Bagehot